#5

Patent Application

of

KLAUS HÖLLIG
ULRICH REIF
and
JOACHIM WIPPER

for

PROCESS FOR INCREASING THE EFFICIENCY OF A COMPUTER IN FINITE ELEMENT SIMULATIONS AND A COMPUTER FOR PERFORMING THAT PROCESS

Field of the Invention

The present invention relates to a process for increasing the efficiency of a computer in finite element simulations by efficient automatic construction of suitable basis functions for computation of approximate solutions, and to a computer for performing that process.

Background of the Invention

A plurality of technical and physical phenomena can be described by partial differential equations. They include among others problems from fluid mechanics (for example, flow around an airfoil), electromagnetic field theory (for example,

electrical field behavior in a transistor) or elasticity theory (for example, deformation of a car body). Accurate knowledge and description of such processes are a central element in the construction and optimization of technical objects. To save time-consuming and cost-intensive experiments, there is great interest in computer-aided simulations. Finite element processes (FE processes) have become established and have been the topic of intense research for a long time. This also applies to automatic mesh generation processes as a foundation for construction of suitable basis functions.

Figure 1 illustrates in the left half the prior art in the process of FE simulation for a linear boundary value problem as a typical model example. Proceeding from the data describing the geometry of the technical object to be simulated, first a system of basis functions is constructed which on the one hand enables fulfillment of boundary conditions and on the other is suited for approximation of the unknown solution. Then, using these basis functions a system of linear equations is set up by numerical integration methods. Finally, the coefficients of the unknown approximation are determined as the solution of this system of linear equations.

FE methods or their use are the subject matter of a series of patents. For example, U.S. Patent No. 4,819,161 discloses a system where FE approximations of a large class of differential equations are automated. U.S. Patent No. 5,731,817 discloses a process for generation of hexahedral meshes forming the foundation for a FE simulation process.

In most FE processes of practical relevance basis functions are used which are defined on a decomposition produced by generating a mesh of the simulation region. Figure 2a shows a selection of conventional elements; their dimension, degree, smoothness and parameters are listed in Figure 2b. A survey of meshing methods of planar regions can be found for example in K. Ho-Le, Finite Element Mesh Generation Methods: A review and classification. Com. Aided Design 20

(1988), 27 – 38. Generating a mesh for complicated three-dimensional regions is extremely difficult using the current state of knowledge, as shown by S. Owen, A survey of unstructured Mesh Generation Technology, Proceedings, 7th International Meshing Round Table, Sandia National Lab (1998), 239 – 257. The processes require extensive amounts of computer time and are to some extent not yet completely automated. But recently, there has been a series of very innovative new approaches. For example, A. Fuchs, Optimierte Delaunay-Triangulierungen zur Vernetzung getrimmter NURBS-Körper, University of Stuttgart, 1999, simulates a force distribution in order to achieve an optimum distribution of triangulation points. In U.S. Patent No. 5, 729, 670 two- and three-dimensional meshes are produced by solving flow problems; this is an interesting reversal of the conventional FE mechanism. In addition, many algorithms have been developed to improve individual aspects of mesh generation processes. For example DE 196 21 434 A1 and U.S. Patent No. 5, 774, 696 describe a process for elimination of intersections with prescribed edges or boundary surfaces in Delaunay triangulations.

Meshless FE-methods to date have not acquired any importance for applications. Both in the Lagrange multiplier method (see for example J.H. Bramble, The Lagrange Multiplier Method for Dirichlet's Problem, Math. Comp. 37 (1981), 1 – 11), and also in the penalty method (see for example P. Bochev and M. Gunzburger, Finite Element Methods of Least Squares Type, SIAM review 40 (1998), 789 – 837), the treatment of boundary conditions represents a major problem in the use of simple, stable basis functions.

In many technical simulations, automatic mesh generation is very complex and requires by far the largest part of the computer time. Furthermore, the approximation power of the conventionally used linear and multilinear basis functions is low. To achieve accurate results, a large number of basis functions must be used, and thus, a correspondingly large system of equations must be

solved. Higher order trial functions on triangulations generally likewise have an unfavorable ratio between the attainable accuracy and the number of basis functions used. Finally, smooth basis functions cannot be easily defined on unstructured meshes. Very special constructions are necessary already for continuously differentiable elements (see Figure 2a).

Summary of the Invention

The object of the present invention is to increase the efficiency of known FE methods and computers which carry out FE methods by efficient construction of basis functions with favorable properties. In particular the meshing of the simulation region will be completely eliminated, optional boundary conditions are fulfilled, accurate solutions are obtained with relatively few coefficients, and the resulting system of equations will be solvable efficiently. In this way the disadvantages of the prior art will be overcome, and thus, the accuracy and speed of the simulation of physical properties in the engineering and optimization of technical objects will be improved.

Some central terms and notations which are used in the following description of the process as claimed in the invention will be explained first.

The simulation region Ω is a bounded set of dimension $d = 2$ or $d = 3$ on which the physical quantities to be studied will be approximated by means of FE methods. The boundary of the simulation region is denoted by Γ. A grid with grid width h is defined as a decomposition of a subset of the plane or the space in grid cells Z_k. Depending on the dimension d each grid cell is a square or a cube with edge length h. More precisely, $Z_k = kh + [0, h]^d$, where k belongs to a set of integer d-vectors. The uniform tensor product B-splines in d variables of degree n with grid width h are denoted by b_k, see for example C. de Boor, A Practical Guide to Splines, Springer, 1978. They are functions which can be continuously differentiated $(n - 1)$ times and which on the grid cells agree with polynomials of degree n, as shown in

Figures 4a and 4b. Figure 4a shows the support Q_k of the B-spline of degree $n = 2$, dimension $d = 2$ and smoothness $m = 1$. In Figure 4b, the resulting tensor product B-spline b_k is shown. The support Q_k, i.e. the union of all grid cells, on which the B-spline b_k is not identical to zero, consists of $(n+1)^d$ grid cells; more precisely, $Q_k = kh + [0, (n+1)h]^d$. In all figures, the B-spline b_k is marked at the point kh, i.e. for example in the case $d = 2$ at the lower left corner of the support. For FE simulations only those B-splines are important which have support intersecting the simulation region Ω; they are called relevant B-splines. The relevant B-splines are again divided into two groups; those B-splines for which the part of the support inside the simulation region is larger than a prescribed bound s are called inner B-splines. All other relevant B-splines are called outer B-splines.

The object of the present invention is achieved by the process defined in claim 1. Special embodiments of the invention are defined in the dependent claims. Claim 11 defines a computer system as claimed in the invention.

The right half of Figure 1 shows the incorporation of the process of the present invention into the course of a FE simulation in the prior art and the substitution of certain process steps of a FE simulation in the prior art by the process of the present invention.

Input 1 of the simulation region Ω can be done via input devices, in particular also by storage of data derived from computer-aided engineering (CAD/CAM). For example, the data used in the engineering of a motor vehicle can be incorporated directly into the FE simulation in the present invention.

In input 2 and storage of the type of boundary conditions natural and essential boundary conditions are distinguished. The basis in the present invention is constructed for homogeneous boundary conditions of the same type. In particular, for essential boundary conditions, the basis functions vanish on the boundary Γ.

Inhomogeneous boundary conditions can be treated in the assembly of the FE systems using methods which correspond to the prior art.

Finally, the control parameters are read in 3. They relate to the degree n and the grid width h of the B-splines to be used and the bound s for classification of the inner and outer B-splines. If specifications are omitted, all these input parameters can be automatically determined by evaluation of merit functions which are constructed empirically or analytically.

The following construction of the basis functions of the invention is divided into the steps shown schematically in Figure 3, which will now be described.

After reading in the simulation region Ω, in the first process step a grid covering the simulation region Ω is generated. Then it is checked which of the grid cells lie entirely inside, partially inside or not inside the simulation region Ω. The cell types 4 are determined, and this information about the cell types is stored. This essentially requires inside/outside tests and determinations of intersections between the boundary Γ of the simulation region Ω and the segments or squares which bound the grid cells. Figure 7 shows the input and output data for this process step.

In the second process step, using the information about the cell types, the relevant B-splines are first determined. Then the classification 5 into outer B-splines is performed; the corresponding lists of indices are denoted by I and J. To this end, the size of those parts of the supports of the B-splines which lie within the simulation region is determined using the data obtained in the first process step, and compared with the prescribed bound s. Figure 9 shows the input and output data for this process step.

In the third process step, coupling coefficients $e_{i,j}$ are computed 6; they join the

inner and outer B-splines according to the rule

$$B_i(x) = b_i(x) + \sum_{j \in J(i)} e_{i,j} b_j(x), \quad i \in I. \tag{1}$$

Hence, an extended B-spline B_i is assigned to each inner B-spline B_i. The construction and the properties of the index sets $J(i)$ and the coupling coefficients $e_{i,j}$ are given as follows. The index sets $J(i)$ consist of indices of outer B-splines. They correspond to complementary index sets $I(j)$ of indices of inner B-splines, i.e., i belongs to $I(j)$ if and only if j belongs to $J(i)$. For a given outer index j the index set $I(j)$ is an array , i.e., a quadratic or cubical arrangement of $(n+1)^d$ inner indices which is characterized by a minimum distance to the index j. For a given outer index j and an inner index i in the index set $I(j)$, let p_i be the d-variate polynomial of degree n in each variable which has the value 1 at the point i and at all other points of the array $I(j)$ the value 0. Then the coupling coefficient $e_{i,j}$ is given as the value of p_i at the point j, i.e., $e_{i,j} = p_i(j)$. The specific values of the coupling coefficient can either be tabulated for different degrees and relative positions of j and $I(j)$ or can be easily computed using Lagrange polynomials. Figure 11 shows the input and output data for this process step.

If natural boundary conditions are given, the extended splines defined in equation (1) are used without modification for further implementation of the FE process. If, on the other hand, essential boundary conditions are given, a weighting according to the rule

$$B_i(x) \leftarrow \frac{w(x)}{w(x_i)} B_i(x), \quad i \in I \tag{2}$$

has to be performed. The pertinent interrogation takes place in an optional process step 6a. The functions defined in this way are called weighted extended B-splines (WEB-splines). Formally, the extended B-splines used under natural boundary conditions correspond to the special case $w(x) = 1$. They are, therefore, also called WEB-splines. For the case of essential boundary conditions, the weight function w is characterized as follows: For all points x of the simulation region, $w(x)$ can be bounded from above and below by positive constants, which are independent of x,

times the distance dist(x) of the point x from the boundary Γ. In other words, w is positive within Ω and tends to zero in the vicinity of the boundary Γ as fast as the distance function dist. For simulation regions which are bounded by elementary geometrical objects (circles, planes, ellipses, etc.) a suitable weight function can optionally be given in explicit analytic form. Otherwise, computation rules should be used which typically represent a smoothing of the distance function. The scaling factor $1/w(x_i)$ is calculated by evaluating the weight function at the weight point x_i. This can be any point in the support of the B-spline b_i which is at least half the bound $s/2$ away from the boundary.

As a result of the process of the present invention, a computation rule for the WEB-splines B_i (compare definitions (1) and (2)) is obtained which have all favorable properties according to the objectives of the invention. Thus the FE method can be continued according to the prior art. But, in doing so, it is possible to advantageously exploit the regular grid structure of the basis functions according to claim 11.

In summary, for the process of the present invention the coupling of outer and inner B-spline is, among other things, important. As a result, the constructed basis has the properties which are essential for FE computations. In particular, a basis B_i (i from the index set I) according to the present invention, is stable, uniformly with respect to the grid width h, and the error has the same order as for the B-splines b_k in the approximation of smooth functions which satisfy the same boundary conditions. On the other hand, the fulfilment of essential boundary conditions is ensured by using the weight function w.

Other objects, advantages and salient features of the present invention will become apparent from the following detailed description which, taken in conjunction with the drawings, listed below, discloses preferred embodiments of the present invention. The features mentioned in the claims and in the specification can be

essential to the invention individually or in any combination.

Brief Description of the Drawings

Referring to the drawings which form a part of this disclosure:

Figure 1 is a flow chart of the individual steps of the prior art in the course of the finite element simulation and integrates the determination of the WEB basis into this process;

Figure 2a compares certain finite elements of the prior art to the WEB element shown in Figure 2b and lists the parameters relevant to finite element approximations;

Figure 3 is a flow chart of the process steps for determining the WEB basis;

Figures 4a and 4b show a support and the corresponding tensor product B-spline of degree 2;

Figures 5a and 5b illustrates the problem formulation of the first embodiment (displacement of a membrane under constant pressure) and of the corresponding solution;

Figure 6 shows the cell types for the first embodiment;

Figure 7 surveys the input and output data of the process for determining the cell types;

Figure 8 illustrates, by way of example, the classification of the B-splines for the first embodiment;

Figure 9 outlines the input and output data of the process for classifying the B-splines;

Figure 10 shows the coupling coefficients of an outer B-spline and the corresponding inner B-splines for the first embodiment;

Figure 11 surveys the input and output data of the process for computing the coupling coefficients;

Figures 12a and 12b illustrates the construction of the weight function of the preferred embodiment;

Figure 13 shows the support of a WEB-spline and the corresponding coupling coefficients for the first embodiment;

Figures 14a and 14b explains the problem formulation of a second embodiment (incompressible flow) and its solution using the flow lines and the distribution of the flow velocity;

Figures 15a to 15c show the changing classification of B-splines for the B-spline degrees $n = 1, 2, 3$ and the same grid width for the second embodiment;

Figures 16a to 16c provide information about the error development in the finite element approximation using WEB-splines and about the computing time behavior of WEB approximations for the second embodiment;

Figures 17a and 17b compare the WEB basis to a process based on linear trial functions on a triangulation (prior art); and

Figure 18 shows a computer system according to the present invention.

Detailed Description of the Invention

One especially favorable embodiment of the process of the present invention, called the WEB process, is determined by the following specifications.

The bound s is chosen such that the inner B-splines are characterized by requiring that at least one of the grid cells of their support lies completely in the simulation region Ω. Since to determine the relevant B-splines, the intersection of the grid cells and the boundary Γ must be computed anyway, the classification requires no significant additional computing time. The weight point x_i is chosen as the midpoint of a grid cell in the support of the B-spline b_i which lies completely in the simulation region Ω. This is also efficiently possible since the determination of one such cell is already part of the classification routine.

If no explicit analytic form of the weight function is known, it is defined by

$$w(x) = \begin{cases} 1 & \text{if } \operatorname{dist}(x) \geq \delta \\ 1 - (1 - \operatorname{dist}(x)/\delta)^n & \text{if } \operatorname{dist}(x) < \delta. \end{cases} \tag{3}$$

Figures 12a and 12b illustrate the construction of the weight function. Here, the parameter δ indicates the width of the strip Ω_δ within which the weight function varies between the value 0 of the boundary of the simulation region and the value 1 on the plateau on $\Omega\backslash\Omega_\delta$. The parameter δ is chosen such that the smoothness of the weight function is ensured.

One important advantage of the process is that no meshing of the simulation region is necessary. In technical applications, this results in clear savings of computing time and storage capacity and simplifies the course of the simulation. The process structure for two- and three-dimensional problems is formally and technically largely identical. This enables time- and cost-saving implementations of solutions for diverse applications based on uniform program structures. The use of B-splines corresponds to the industrial standard in the modeling of geometrical

objects, and thus forms a natural connection between FE and CAD/CAM applications. Extensive existing program libraries from both fields can be used for implementing a FE simulation based on the process of the present invention. The basis functions constructed using the WEB process have all standard properties of finite elements. This includes especially the stability of the basis. It implies, for example, that for linear elliptic boundary value problems the condition number of the resulting system of equations does not grow faster than for optimal triangulations as the grid width becomes smaller. For applications, this means, for example, that linear systems of equations as they typically arise in FE methods, can be efficiently solved by iterative algorithms. Furthermore, for a given degree, the approximation order is maximal and the number of necessary parameters minimal. Thus, very accurate approximations are possible with a relatively small number of parameters. Specifically, this can mean that the accuracies which so far required the use of mainframe computers can now be achieved with workstations. The regular grid structure of the basis of the present invention permits a very efficient implementation, especially for assembling and solving FE systems. Moreover, by using the weight function, the boundary conditions can be satisfied during simulation without affecting the regular grid structure of the basis functions. Finally, by using multigrid methods to solve the linear systems arising in linear elliptic boundary value problems, one can achieve that the overall solution time is proportional to the number of unknown coefficients, and thus optimal.

The process of the present invention in the special preferred embodiment (WEB process) is illustrated using the first embodiment shown in Figures 5a and 5b. The differential equation and boundary conditions are chosen to be very elementary so that in addition to the construction of the WEB basis of the present invention the entire course of the FE simulation can be followed without major additional effort.

Figure 5a shows an elastic membrane, which is fixed along the edge Γ of a planar simulation region Ω, and on which a constant pressure $f = 1$ acts inside the region.

With suitable normalization, the displacement u satisfies the Poisson equation with homogeneous boundary conditions,

$$\begin{aligned} -\Delta u &= 1 \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma. \end{aligned}$$

The displacement u or the deflection of the membrane is depicted in Figure 5b. As described above, the WEB process is divided into the following steps.

Input 1 of the simulation region Ω: The boundary Γ is a periodic spline curve of degree 6, which is stored by its control points 20 (in Figure 5a identified with black dots).

Input 2 of the boundary conditions: the homogeneous boundary condition is essential so that the construction of a weight function is necessary.

Input 3 of the control parameters: The degree $n = 2$, and, in order to make the figures easier to understand, a relatively large grid width $h = 1/3$ are used.

Determination 4 of the cell types: As illustrated in Figure 6, the simulation region is covered by a grid 21, which contains the grid cells of the supports of all B-splines of potential relevance for the basis construction. The type determination in the example yields 69 outer grid cells 22, and 11 inner grid cells 24, and 20 grid cells 23 on the boundary.

Classification 5 of the B-splines: Here, the support of the B-spline b_k is a square Q_k with corners

$$(k_1, k_2)h,\ (k_1 + 3, k_2)h,\ (k_1 + 3, k_2 + 3)h,\ (k_1, k_2 + 3)h;$$

$Q_{(-1,0)}$ and $Q_{(2,1)}$ are shown in Figure 8. The grid points kh of the relevant B-splines, for which Q_k intersects the interior of the simulation region, are marked

in Figure 8 by a point or a circle. All grid points ih for inner B-splines (i from the index list I), for which at least one cell of the support Q_i lies entirely within Ω, are marked by a point. For example, for $i = (-4, 0)$, the grid cell $(-2, 0)h + [0, h]^2$ lies entirely in Ω. All grid points jh for outer B-splines (j from the index list J), for which no cell of the support Q_j lies entirely in Ω, are marked by a circle.

Computation 6 of the coupling coefficients: To determine the coupling coefficients $e_{i,j}$ for each fixed j of the index list J the nearest 3×3-array

$$I(j) = \{\ell_1, \ell_1 + 1, \ell_1 + 2\} \times \{\ell_2, \ell_2 + 1, \ell_2 + 2\}$$

of indices in I is sought. In Figure 10, for the outer grid point $j = (-1, 2)$, which is marked with a circle, the array $I(j)$ is identified with points. Figure 10 likewise shows the corresponding coupling coefficients in a matrix representation. They are computed by bivariate interpolation. For example, the interpolating polynomial for $i = (-1, -1)$ is

$$p_i(x) = -x_1(x_1 + 2)x_2(x_2 - 1)/2.$$

Its value at the point $x = j = (-1, 2)$ is $p_i(j) = 1 = e_{i,j}$. One notices that many of the coupling coefficients are 0. This is a typical phenomenon. The coupling coefficients $e_{i,j}$ are not equal to 0 for all i of the index list $I(j)$ only if the indices i are different from the index j in each component.

Computation rule 7 for the weight function: The weight function is given by equation (3) with $n = 2$ and $\delta = 0.2$. The parameter δ is computed numerically. It must be small enough to prevent singularities of the distance function. To compute the distance function, the process generates a program which uses Newton's method. Since the weight function is not equal to 1 only in a boundary strip, the complexity in the subsequent evaluations is low.

Output: Figure 13 shows the support of a WEB-spline B_i and the data necessary for its description. These are the index list $J(i)$ of the outer B-splines b_j coupled

with b_i, the coupling coefficients $c_{i,j}$, and the weight point x_i. These data are used in conjunction with the weight function for generating the computation rule for the WEB-splines.

The further course of the FE simulation follows the prior art.

Assembly 9 of the FE system: The entries of the system matrix and of the right-hand side are

$$G_{k,i} = \int_{\Omega} \operatorname{grad} B_k \operatorname{grad} B_i, \; F_k = \int_{\Omega} f B_k, \quad k, i \in I.$$

The system of equations $GC = F$ for the basis coefficients C_i in this example has dimension 31. The matrix entries $G_{i,j}$ are computed using numerical integration, likewise the integrals F_k.

Solution 10 of the FE system: The Galerkin system is solved iteratively with the conjugate gradient method with SSOR preconditioning used to accelerate convergence. After 24 iteration steps the solution is found within machine accuracy (tolerance $\leq 1e - 14$).

Computation 11 and output 12 of the approximation: The approximation computed with the process as claimed in the invention is $u = \sum_i C_i B_i$ and is shown graphically in Figure 5b. The relative error of the L_2-norm is 0.028.

The efficiency of the process of the present invention in the special preferred embodiment (WEB process) is illustrated in a second embodiment using the simulation of an incompressible flow. The arrangement of two circular obstacles shown in Figure 14a in a channel with parallel boundaries serves to illustrate the principal strategy. For complicated geometries, as are typically present in specific applications, the process works completely analogously and efficiently. In Figure 14a the stream lines 25 are shown within the region bounded by Γ_1 to Γ_4

and by Γ_5 and Γ_6. The differential equation is:

$$\Delta u = 0 \qquad \text{in } \Omega$$

with the boundary conditions

$$\frac{\partial u}{\partial n} = v_0 \ \text{ on } \Gamma_1\,, \qquad \frac{\partial u}{\partial n} = -v_0 \ \text{ on } \Gamma_2\,, \qquad \frac{\partial u}{\partial n} = 0 \ \text{ on } \Gamma_3, \ldots, \Gamma_6.$$

The flow velocity $v = -\operatorname{grad} u$ is shown in the bottom half of the figure.

The construction of the WEB basis of the present invention proceeds completely analogously to the first embodiment. The sole difference is that a weight function is not necessary because of the natural boundary conditions.

Figures 15a to 15c show the classification of the relevant B-splines for different degrees n (see also Figure 8). In the figure, the inner B-splines b_i, which are taken into the WEB basis without extension, are marked by solid triangles. For small h the number of these B-splines increases, i.e., $B_i = b_i$ for most of the WEB basis. For degree $n = 3$ this is the case for 236 of 252 indices in the example.

Figure 16a shows in two diagrams the numerically determined relative L_2-error of the potential (left half of the figure) as a function of the grid width $h = 2^{-k}$ with $k = 1, \ldots, 5$ and the numerically estimated order of convergence m (right half of the figure). Here, for different degrees of the WEB-spline the following markers are used: $*$ ($n = 1$), $\circ$ ($n = 2$), $\triangle$ ($n = 3$), $\square$ ($n = 4$) and $\star$ ($n = 5$). As expected, $m \approx n + 1$, i.e., an approximate error reduction by a factor 2^{n+1} when the grid width is cut in half. Analogously, for the relative approximation error of the flow velocity shown in Figure 16b (H^1-norm of the solution, left half of the figure), an order of convergence $m \approx n$ (right half of figure) is obtained with an associated error reduction by roughly a factor 2^n when the grid width is cut in half.

Figure 16c (right half of the figure) shows the computing time in seconds for construction of the WEB basis as a function of the number of resulting basis

functions, measured on a Pentium II processor with 400 MHz. For example, for construction of a WEB basis of degree 3 with grid width $h = 0.125$ with 2726 WEB-splines 1.32 seconds are necessary. One notices that the complexity for generating the WEB basis is largely independent of the degree n of the basis. In the left half of Figure 16c the number of CG-iterations relative to the number of basis functions is shown. Thus, for the corresponding system with 2726 unknowns, 65 PCG-iterations are required. The total computing time including assembling and solving the Galerkin system is roughly 2.48 seconds.

Figures 17a and 17b compare the WEB process with a standard solution process which meshes or triangulates the simulation region (Fig. 17a) and uses hat functions. The graph shows in Fig. 17b the L_2-error relative to the number of parameters. The results of the standard solver are marked with boldfaced diamonds and are compared to the results achieved using the WEB basis of degrees 1 to 5. For example, an accuracy of 10^{-2} is achieved with the WEB process by using 213 basis functions with degree 2 and an overall computer time of 0.6 seconds. To achieve the same accuracy, the standard method with linear hat functions required 6657 basis functions.

In the assessment of the standard solution process two other aspects must be considered. On the other hand, Figure 17b illustrates that even a moderate accuracy of 10^{-3} can only be achieved with hat functions when far more than one million coefficients are used. This shows that when using hat functions accurate results generally require an enormous computing and storage capacity or cannot be achieved at all with the prior art. On the other hand, the complexity required for meshing increases with the complexity of the simulation region. In contrast to realistic applications, the region studied here is comparatively simple to triangulate due to its simple structure.

The two-dimensional example shows the performance gain by the WEB process.

An even greater increase in performance is possible in three-dimensional problems. On the one hand, the complexity for meshing, which is eliminated in the WEB process, is much greater. On the other hand, the reduction in the number of required basis functions becomes much more noticeable than in the two-dimensional case.

Figure 18 shows a device according to the present invention, especially a computer system 30, with input devices 31, 32, 33, output devices 34 and a control unit 35 which controls the course of the process. To carry out the process of the present invention and in particular for purposes of parallelization of the pertinent computations, the central control unit 35 preferably uses several arithmetic logic units (ALU) or even several central processing units (CPU) 36. These allow especially parallel processing for the process steps classification 5 of the B-splines, in particular also intersection of the regular grid with the simulation region Ω, determination 6 of the coupling coefficients $e_{i,j}$, and/or evaluation of the weight function $w(x)$ at points x of the simulation region Ω.

The computer units 36 here access the common data resources of the storage unit 37. The data can be input, for example, by a keyboard 31, a machine-readable data medium 38 via a corresponding read station 32 and/or via a wire or wireless data network with a receiver station 33. Via the read station 32 or a pertinent data medium 38, the control program, which controls the process execution, can be input, and, for example, can be permanently filed on the storage media 37. Accordingly, the output devices 34 can be a printer, a monitor, a write station for a machine-readable data medium and/or the transmitting station of a wire or wireless data network.

While various embodiments have been chosen to illustrate the invention, it will be understood by those skilled in the art that various changes and modifications can be made therein without departing from the scope of the invention as defined in